**REMARKS**

The present amendment is responsive to the Office Action dated March 4, 2008. Claims 1, 4, 6-7, 13, 18-20, 26, 32-33 and 43-45 have been amended. No new matter has been introduced by these amendments. Claims 2, 5, 8 and 27 have been cancelled. Thus, claims 1, 3-4, 6-7, 9-26 and 28-50 are again presented for consideration in view of the following remarks. The rejections will be addressed in view of the claims as presented herein.

As an initial matter, claim 44 was objected to due to a typographical error. That error has been corrected. Thus, applicants respectfully request that the objection to claim 44 be withdrawn.

Claims 1-3 were provisionally rejected on the ground of obviousness-type double patenting over claims 10 and 12 of copending U.S. Application No. 10/783,238. A terminal disclaimer is submitted concurrently herewith. In view of this, it is requested that the obviousness-type double patenting rejection be withdrawn.

Claims 1-50 were rejected under 35 U.S.C. § 112, second paragraph as being indefinite. Applicants respectfully traverse the rejection. Of these, claims 1, 4, 19, 26, 29 and 43 are independent.

The Office Action raised three indefiniteness questions regarding claim 1 and equivalent questions as to claims 4, 19, 26, 29 and 43. (*See* Office Action, p.4, numbered sections 9.a.i and 9.a.ii)

First, the Examiner asked if "processor tasks" is meant to be tasks "to be executed on a specific processor," tasks "that are waiting in the queue to be executed," or tasks "that are being executed." (Office Action, p.4, numbered section 9.a.i.) Second, the Examiner asked where processor tasks reside. (*See Id.*) Third, the Examiner asserted that the meaning of "may access" was unclear as to "whether or not the processing units access the shared memory." (*Id.*)

In addressing the first and the second questions, applicants refer to the teachings of the specification. The term "processor tasks" refers to the processing instructions and/or the processing data in a software application. (*See* specification ¶ 0004, ll.1-5.) A processor task may reside in a software application and is stored in the shared memory of the multi-processor system, where it may also be copied from the shared memory to the local memory of a sub-processor. (*See* specification ¶ 0067, ll.6-17 and ¶ 0088, ll.7-10.)

The third question has been resolved by clarifying the second clause of claim 1 to recite "storing the processor tasks in a shared memory that is accessible by a plurality of processing units of the multi-processor computing system." Claims 4 and 19 have also been amended to change "may be accessed" to "is accessible." Original claims 26, 29 and 43 did not include the "may be accessed" language.

Claim 2 was rejected under § 112, ¶ 2 because the recitation "copied from the shared memory" purportedly had an unclear meaning as to "where it is being copied to from the shared memory." (Office Action, p.4, numbered section 9.a.iii.) Claims 5, 8, 27 and 33 were similarly rejected. (*Id.* at section 9.a.iv.) Claims 2, 5, 8 and 27 have been cancelled, thereby mooting their rejections. However, aspects of claim 2 have been incorporated into claim 1, aspects of claims 5 and 8 have been incorporated into claim 4, and aspects of claim 27 have been incorporated into claim 26.

Each of independent claims 1, 4 and 26 have been amended to clarify where copying has been done from and to. With regard to claim 33, this claim depends from independent claim 29. Claim 29 is clear as to where the processor tasks should be copied from and to. Specifically, claim 29 recites "wherein the processing units are operable to use the task table to determine which of the processor tasks should be copied from the shared memory into their local memories and executed." Thus, applicant submit that it is clear that in claim 33 copying occurs between the shared memory and the local memory of the processors.

Claim 18 was rejected on a similar ground as claims 2, 5, 8, 27 and 33. (*See* Office Action at p.5, numbered section 9.x.) Claim 18 has been amended to clarify where the copying is done to. Support for such amendment may be found at specification paragraph 0088, ll.11-12 and paragraph 0089, ll.18-19.)

Claims 7, 20, 32 and 45 were rejected on two different grounds. (*See* Office Action at p.5, numbered sections 9.v and 9.vi.) The first ground was that "it is unclear how 'task table' and 'task queue' related to each other and what is the difference between them in terms of task listing." Applicants would refer to the specification, which clearly explains and discusses such terms. For instance, the term "task table" is explained in details by the specification at paragraphs 0081-0084 and is conceptually illustrated in FIGS. 5 and 6.

In one embodiment, the task table preferably includes a plurality of task table entries and the task table has at least one linked list of task table entries. In such a case as explained in the specification, each task table entry is associated with one of the processor tasks and each entry has at least one of four indicators (status indication STAT, priority indication PRI, a pair of pointers pointing to the previous task PREV and the next task NEXT.)

The term "task queue" is also explained in full detail by the specification and is conceptually illustrated in FIG. 7. As shown and described, the task queue is used to aid in the sub-processor's identification of the first (or head) entry of the linked list stored in the task table. In an embodiment in the specification, the queue preferably includes an entry for each priority level of the associated processor tasks and each entry includes at least one of a HEAD pointer and a TAIL pointer. (*See* specification paragraph 0085 and FIG. 7.) As explained, the sub-processor utilizes both the task table and the task queue to determine which of the processor tasks should be copied from the shared memory for execution and to execute the processor tasks in the proper order. *See* specification paragraph 0087, ll.10-17, as well as FIGS. 8-10.) The sub-processor also maintains and modifies the task table and the task queue during the execution of the processor tasks. *See* specification paragraph 0088, ll.1-7, as well as FIGS. 8-10.)

The second ground on which claims 7, 20, 32 and 45 were rejected was that the term "the list" purportedly has unclear meaning and the Examiner did not understand "which list is used to execute the task." (Office Action, p.5.) Claim 13 was similarly rejected. (*See* Office Action, p.5, numbered section ix.) Applicants submit that the meaning of the term "the list" is clear, as each of these claims depends from a prior claim that defines a list of processor tasks. For instance, the term "the list" in claim 7 refers to "the list of processor tasks" in claim 4.

Nonetheless, in order to advance prosecution, claims 7, 13, 20, 32 and 45 have been amended to clarify that it is the list of processor tasks. This amendment does not alter the scope of the term in these claims.

In rejecting claim 11, the Examiner questioned the meaning of "copying task queue and table" and asked "if the entire task table and queue is being copied or only the entry related to the processor and why both are being copied."

(Office Action, p.5, numbered section 9.vii.) Claims 21, 36 and 46 were similarly rejected. (*Id.* at section 9.viii.)

Applicants first refer to the previous discussion for claim 1 regarding the meaning of "processor task," and the previous analysis for claim 7 regarding the differences between "task table," "task queue" and their relationship.

Applicants respectfully submit that in view of the specification and the figures, it is clear that the task table, task queue and the processor tasks to be executed are all copied from the shared memory into the local memory. "[A] particular sub-processing unit 102 is called to initiate the copying of **a processor task** 110 from the shared memory 106 to the local memory thereof. (Specification paragraph 0088, ll.7-10, emphasis added; *see also* Fig. 8.) "[T]he sub-processing unit 102 locks and copies the **task queue** 282 into its local memory." (Specification paragraph 0088, ll.10-12, emphasis added.) "Thus, at action 308 that **task table** is labeled and copied into the local memory of the SPU 102 such that it may be modified." (Specification paragraph 0088, ll.35-37, emphasis added.)

In view of the above, applicants submit that all claim terms at issue are definite and respectfully request that the § 112, §2 rejections be withdrawn.

Claims 1-6 and 26-31 were rejected under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 5,592,671 ("*Hirayama*"). Of these, claims 1, 4, 26 and 29 are independent. Applicants respectfully traverse the rejection.

Independent claims 1, 4 and 26 have been amended as recited above. Applicants submit that *Hirayama* does not teach or otherwise suggest each and every limitation of these independent claims.

Applicants respectfully disagree with the Office Action's assertions as to claim 29. Here, the Office Action contends that *Hirayama* teaches that the processing units "determine which of the processor tasks should be copied from the shared memory into their local memories and executed (col 2, lines 54-62; lines 64-67; col 5, lines 33-37; col 6, lines 9-16; lines 37-40)." (Office Action, p.8, numbered section 18.) Applicants respectfully disagree. What *Hirayama* states is:

> FIG. 1 is a block diagram of a resource management system of the first embodiment. Processors 100, 101, 102, . . . , 10n are connected to a common memory (or a shared memory) 12 via a bus 11, thereby forming a UMA (Uniform Memory Access)-type multiprocessor system, or a tightly coupled multiprocessor system. A process management table 13 contained in the common memory 12 is made up of process management table entries and is used to manage processes executed on the system. The individual process management table entries 130, 131, . . . , 13m are used to manage corresponding processes 140, 141, . . . , 14m, respectively. When there are executable processes on the system, those processes are linked to a run queue 15. Each processor selects a process to be executed from the processes linked to the run queue 15.

(*Hirayama*, col. 2, ll.54-62, 64-67.)

> The values of those priorities are expressed as high or low. For the initially set value for the priority, the value of the priority corresponding to a page on a local memory of a processor is made high, and the value of the priority corresponding to a page on a remote memory is made low.

(*Hirayama*, col. 5, ll.33-37.)

> Because the individual processors 500, 501, 502, . . . , 50n, when allocating pages to processes, select pages in the order of descending priorities, if processes move less frequently between processors, a processor will access a local memory more frequently, thereby enabling an efficient memory management. To cause processes to move less frequently between processors, a process management as explained in the first embodiment is effected.

(*Hirayama*, col. 6, ll.9-16.)

> If such a page is not present, the processor will call the page out demon and repeat step C1 (step C7). If such a page is present, the processor will allocate the page to a process. At step C5, if the value of the priority is high, the page being checked will also be allocated to a process (step C8).

(*Hirayama*, col. 6, ll.37-40.)

Applicants submit that none of the cited portions of *Hirayama* actually teaches what the Office Action asserts it teaches. The first cited portion describes the general block diagram of the resource management system, the process management table, the entries therein, and the run queue. The other cited portions describe how the system prioritizes and allocates the pages of the processor's local memory.

In contrast, claim 29 recites that the processing units are operable to use the task table to determine which of the processor tasks should be copied from the shared memory into the local memories and be executed. This is not what *Hirayama* teaches.

In view of the above, applicants submit that independent claims 1, 4, 26 and 29 are patentable over *Hirayama*. Furthermore, claims 3, 6, 28 and 30-31 depend from claims 1, 4, 26 and 29, respectively, and contain all the limitations thereof. For at least this reason, the subject dependent claims are likewise patentable over *Hirayama*.

Claims 7-14, 16-22, 32-39 and 41-47 were rejected under 35 U.S.C. § 103(a) as being unpatentable over *Hirayama* in view of EP 0459931 ("*Bahr*"). And claims 15, 23-25, 40, 48-50 were rejected under 35 U.S.C. § 103(a) as being unpatentable over *Hirayama* in view of U.S. Patent No. 6,321,308 ("*Arnon*"). Of these, claims 19 and 43 are independent. Applicants respectfully traverse the rejections.

As claims 7, 9-18 and 32-42 are dependent from claims 4 and 29, respectively, and contain all the limitations thereof, applicants submit that these subject dependent claims are patentable for at least this reason.

Independent claims 19 and 43 have been amended to clarify use of the tasks tables. For instance, claim 19 now recites, in part, "modifying the task table entry for the first one of the processor tasks that yields the given processing unit; and writing the task table back from local memory of the given processing unit to the shared memory." And claim 43 now recites, in part, "the sub-processing units are operable to modify the task table entry for the first one of the processor tasks that yields a given processing unit, and the sub-processing units are operable to write the task table back from the local memory of the given processing unit to the shared memory." Support for the amendments may be found, by way of example only, at specification paragraph 0090 and FIG. 9.

In rejecting claims 19 and 43, the Office Action relied upon *Bahr* for the proposition that it purportedly "teaches at least initiating execution a first one of the processor tasks of the list within a given one of the processing units, wherein the first one of the processor tasks yields the given processing unit such that it is capable of executing another of the processor tasks (col 1, lines 24-28)." (Office Action, p.13, numbered section 37.)

However, applicants submit that *Bahr* does not teach that the processing unit modifies the status parameter of the task table entry for the first one of the processor tasks that yields the given processing unit to another processor tasks and then writes the task table back from the local memory to the shared memory as claimed. Thus, for at least this reason, applicants submit that independent claims 19 and 43 are patentable over *Hirayama* and *Bahr* in the asserted combination.

Furthermore, claims 20-25 and 44-50 depend from claims 19 and 43, respectively, and contain all the limitations thereof. In view of this, applicants submit that these subject dependent claims are patentable for at least this reason.

Notwithstanding the patentability of the dependent claims due to their dependency on the allowable independent claims, applicants submit that the dependent claims are separately patentable in view of the limitations contained therein.

By way of example only, claim 17 is independently patentable notwithstanding its dependency from claim 4. Claim 17 was rejected on the ground that *Bahr* "teaches copying the task queue and the task table from the local memory of the given sub-processing unit into the shared memory when the given sub-processing unit has completed its use thereof (col 10, lines 41-49)." (Office Action, p.12, numbered section 35.) However, what *Bahr* actually states is:

> At the beginning of the reverse scan, the Assigned Task Count is compared with a "usable" processing device count (Usable Count) equal to the lesser of (i) the number of processing devices and (ii) the number of tasks in the task dispatching queue. If, in view of the comparison at 140, the Assigned Task Count equals the Usable Count, the allocation of tasks is complete, and the tasks are dispatched, each to its associated one of processing devices 18-24.

(*Bahr*, col. 10, ll.41-49.)

The cited portion of *Bahr* refers to a reverse scan stage that scans the data operations in the order of ascending priority. (*See also Bahr*, col. 4, ll.7-11.) However, this is not what is claimed and has nothing to do with copying the task queue and the task table from the local memory of the given sub-processing unit into the shared memory. Thus, applicants submit that claim 17 is patentable by itself notwithstanding the patentability of claim 4.

As it is believed that all of the rejections set forth in the Office Action have been fully met, favorable reconsideration and allowance are earnestly solicited.

If, however, for any reason the Examiner does not believe that such action can be taken at this time, it is respectfully requested that he telephone applicants' attorney at (908) 654-5000 in order to overcome any additional objections which he might have. If there are any additional charges in connection with this requested amendment, the Examiner is authorized to charge Deposit Account No. 12-1095 therefor.

Dated: September 2, 2008

Respectfully submitted,
Electronic signature:
/Andrew T. Zidel/
Andrew T. Zidel
Registration No.: 45,256
LERNER, DAVID, LITTENBERG,
KRUMHOLZ & MENTLIK, LLP
600 South Avenue West
Westfield, New Jersey 07090
(908) 654-5000
Attorney for Applicant

915518_1.DOC